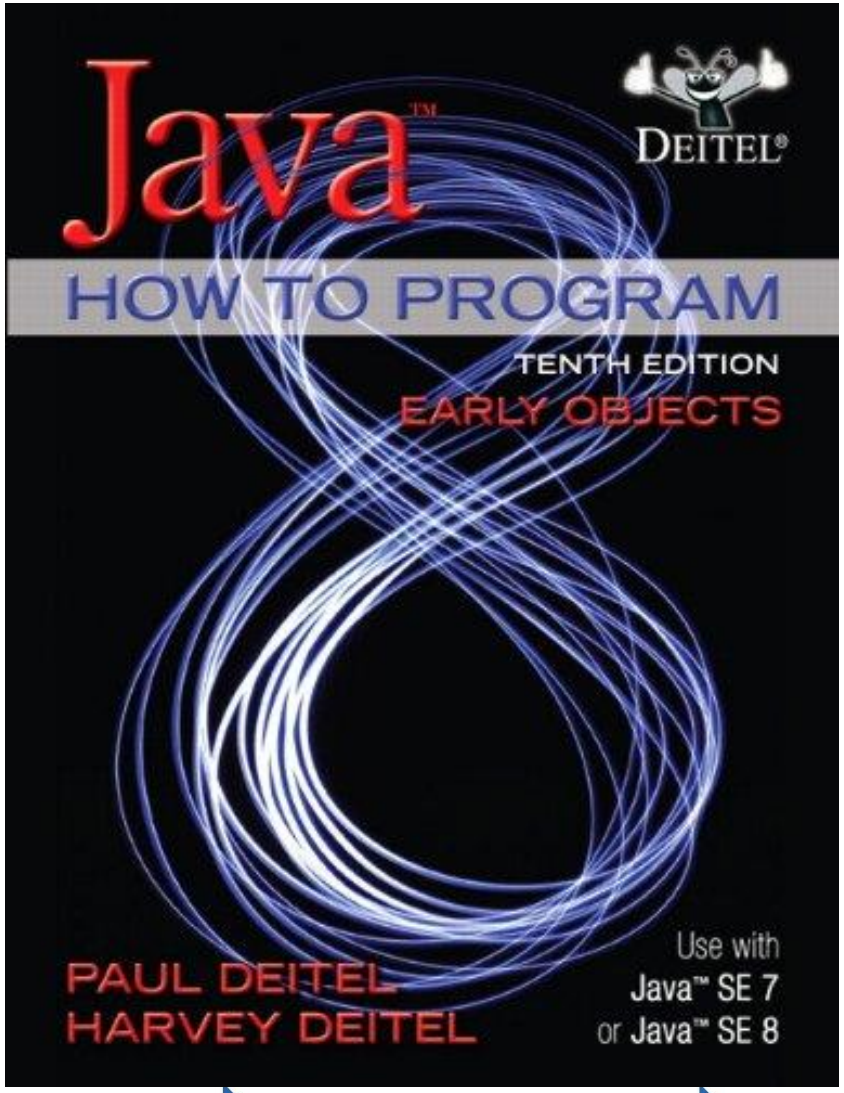
Object Oriented Programming (Java)

Java Arrays

Text Book



Title: Java How to Program, Early Objects

Author(s): Paul Deitel, Harvey Deitel

Publisher: Pearson Education

Year: 2015

ISBN: 0133807800,9780133807806

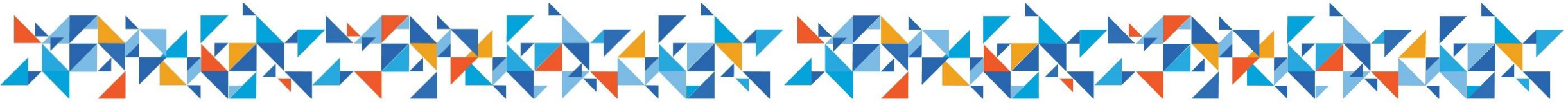
Object Oriented Programming using Java by Simon Kendal

Learning Outcomes

پوهنتون کاردان
KARDAN UNIVERSITY

- **Students will be able to understand**

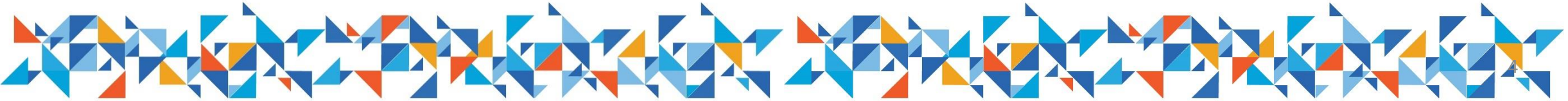
- **Arrays in Java**



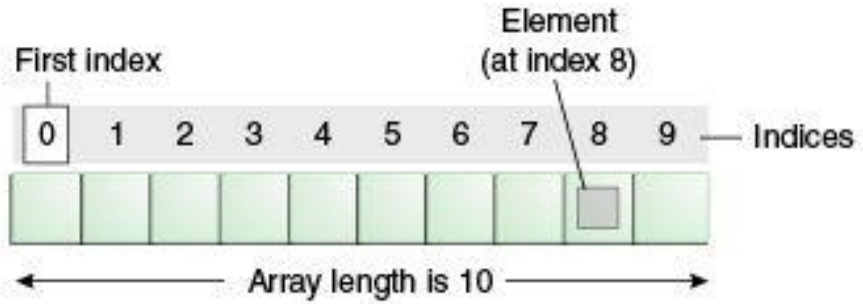
Java Arrays

پوهنتون کاردان
KARDAN UNIVERSITY

- Normally, an array is a collection of similar type of elements which has contiguous memory location.
- **Java array** is an object which contains elements of a similar data type.
- Additionally, The elements of an array are stored in a contiguous memory location.
- It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.
- or
- Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.



- Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.



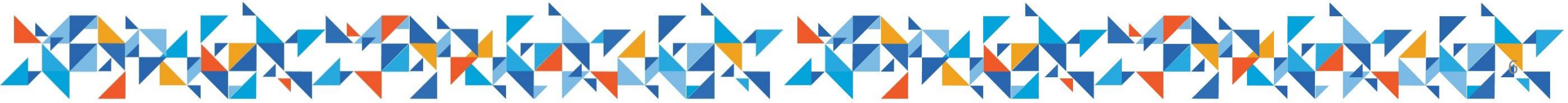
- Unlike C/C++, we can get the length of the array using the length member.
- In C/C++, we need to use the sizeof() operator.

- To declare an array, define the variable type with **square brackets**:

```
String[] cars;
```

- We have now declared a variable that holds an array of strings. To insert values to it, you can place the values in a comma-separated list, inside curly braces:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```



- To create an array of integers, you could write:

```
int[] myNum = {10, 20, 30, 40};
```

Access the Elements of an Array

- You can access an array element by referring to the index number.
- This statement accesses the value of the first element in cars:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
// Outputs Volvo
```

Change an Array Element

- To change the value of a specific element, refer to the index number:

Example

```
cars[0] = "Opel";
```

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
System.out.println(cars[0]);
// Now outputs Opel instead of Volvo
```

Array Length

- To find out how many elements an array has, use the “length” property:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars.length);
// Outputs 4
```

Traversing array using For loop to print all the elements of array

```
public class traversing_array_elements {
    public static void main(String[] args) {

        int array[]={10,20,30,40,50,60,70,80,90};

       for(int i=0; i<array.length; i++){
           System.out.println(array[i]);
        }


    }

}
```

Another way to declare & print all the elements of array

پوهنتون کاردان
KARDAN UNIVERSITY

```
public class traversing_array_elements {
    public static void main(String[] args) {

        int numbers[]=new int[5];
        numbers[0]=10;
        numbers[1]=20;
        numbers[2]=30;
        numbers[3]=40;
        numbers[4]=50;

        for(int i=0; i<numbers.length; i++){
            System.out.println(numbers[i]);
        }


    }
```

For-each Loop for Java Array

- We can also print the Java array using **for-each loop**.
- The Java for-each loop prints the array elements one by one.
- It holds an array element in a variable, then executes the body of the loop.

Syntax

- The syntax of the for-each loop is given below:

```
for(data_type variable:array){
//body of the loop
}
```

Example

```
public class foreachloop {
    public static void main(String[] args) {

        int marks[]={66,77,88,99,100};

        for(int i:marks){
            System.out.println(i);
        }


    }
}
```

Two Dimensional Array

- In Java, a 2D array (two-dimensional array) is essentially an array of arrays. It is commonly used to represent matrices or tables.

Declaring a 2D Array

- dataType[][] arrayName;

- For example:
- int[][] matrix;

Creating (Allocating Memory) for a 2D Array

- arrayName = new dataType[rows][columns];

- Example:
- matrix = new int[3][4]; // 3 rows, 4 columns

Initializing a 2D Array with Values

پوهنتون کاردان
KARDAN UNIVERSITY

- You can initialize it directly like this:

- int[][] matrix = {
- {1, 2, 3},
- {4, 5, 6},
- {7, 8, 9}
- };

Accessing Elements

- Use two indices: one for row and one for column.
- int value = matrix[1][2]; // Access element at 2nd row, 3rd column
- matrix[0][0] = 99; // Set value at 1st row, 1st column

Questions ?

پوهنتون کاردان
KARDAN UNIVERSITY

Thank You…!